

Amendments to the Claims

Please cancel Claims 12, 40, 58, 61 and 63. Please amend Claims 1, 13, 14, 17, 29, 41, 42, 43, 44, 45, 57, 60, 62, and 65. The Claim Listing below will replace all prior versions of the claims in the application:

Claim Listing

1. (Currently Amended) A data processing system for processing programs, the system comprising:
 - memory for storing data objects, the data objects being referenced by pointers;
 - [[and]]
 - a short-quasi-unique-identifier (SQUID) generator which generates SQUIDs for newly allocated data objects to be stored in the memory segment, pointers to a particular data object being associated with the data object's SQUID; and
 - a comparator which compares SQUIDs associated with two different pointers.
2. (Original) The system of Claim 1, further comprising:
 - a memory allocator which allocates a segment of the memory to a data object.
3. (Original) The system of Claim 2, wherein if the data object is moved to a second allocated memory segment, a pointer to the second allocated memory segment is placed at the original memory segment.
4. (Original) The system of Claim 3, wherein the data object is moved due to resizing.
5. (Original) The system of Claim 3, wherein the data object is moved from a first memory to a second memory within a distributed system.
6. (Original) The system of Claim 3, wherein the data object is moved due to garbage collection.

7. (Original) The system of Claim 3, wherein the data object is moved due to data compaction.
8. (Original) The system of Claim 1, wherein the distribution of SQUIDs over a range is uniform.
9. (Original) The system of Claim 8, wherein SQUIDs are generated by counting.
10. (Original) The system of Claim 8, wherein SQUIDs are generated randomly.
11. (Original) The system of Claim 8, wherein SQUIDs are generated by hashing.
12. (Canceled)
13. (Currently Amended) The system of Claim [[12]] 1, further comprising:
an instruction reordering mechanism which reorders instructions and which is responsive to the comparator.
14. (Currently Amended) The system of Claim [[12]] 1, the comparator determining that the two pointers do not reference the same data object if the SQUIDs are different.
15. (Original) The system of Claim 14, the comparator further determining that the two pointers reference the same data object if the SQUIDs are identical and address fields of the two pointers are identical.
16. (Original) The system of Claim 15, each pointer address field comprising:
a base address; and
an offset,
the comparator further determining the two pointers do not reference identical locations within a referenced data object if the pointers' offsets are not identical.

17. (Currently Amended) The system of Claim [[12]] 1, wherein a pointer is associated with a migration indicator field which indicates a number of migrations of the referenced data object by the time said pointer is created, the comparator determining that said two pointers do not reference the same data object if their associated migration indicators indicate identical numbers of migrations and their corresponding addresses are different.
18. (Original) The system of Claim 17, wherein the migration indicator comprises one bit.
19. (Original) The system of Claim 1, wherein at least one pointer is a guarded pointer.
20. (Original) The system of Claim 1, wherein the SQUID is implemented by hardware.
21. (Original) The system of Claim 1, wherein the SQUID is implemented by software.
22. (Original) The system of Claim 1, wherein a pointer contains its associated SQUID.
23. (Original) The system of Claim 1, further comprising a SQUID cache for storing SQUIDS of recently-used pointers.
24. (Original) A data processing system for processing programs, the system comprising:
 - memory for storing data objects;
 - pointers to data objects stored in the memory;
 - migration indicators associated with pointers, a migration indicator indicating a number of migrations of a data object referenced by an associated pointer prior to said pointer being created; and
 - a comparator which determines that said two pointers do not reference the same data object if their associated migration indicators indicate identical numbers of migrations and their corresponding addresses are different.

25. (Previously Presented) The system of Claim 24, wherein the migration indicator comprises one bit.
26. (Previously Presented) The system of Claim 24, wherein the migration indicator comprises multiple bits.
27. (Previously Presented) The system of Claim 24, wherein the migration indicator is implemented by hardware.
28. (Previously Presented) The system of Claim 24, wherein the migration indicator is implemented by software.
29. (Currently Amended) A method for processing programs, the system comprising:
 - storing data objects in a memory, the data objects being referenced by pointers;
 - [[and]]
 - generating a short-quasi-unique-identifier (SQUID) and assigning the SQUID to a data object stored in the memory segment, pointers to the data object being associated with the data object's assigned SQUID; and
 - comparing SQUIDs of two different pointers.
30. (Previously Presented) The method of Claim 29, further comprising:
 - allocating a segment of the memory to the data object.
31. (Previously Presented) The method of Claim 30, wherein if the data object is moved to a second allocated memory segment, a pointer to the second allocated memory segment is placed at the original memory segment.
32. (Previously Presented) The method of Claim 31, wherein the data object is moved due to resizing.

33. (Previously Presented) The method of Claim 31, wherein the data object is moved from a first memory to a second memory within a distributed system.
34. (Previously Presented) The method of Claim 31, wherein the data object is moved due to garbage collection.
35. (Previously Presented) The method of Claim 31, wherein the data object is moved due to data compaction.
36. (Previously Presented) The method of Claim 29, wherein the distribution of SQUIDs over a range is uniform.
37. (Previously Presented) The method of Claim 29, wherein SQUIDs are generated by counting.
38. (Previously Presented) The method of Claim 29, wherein SQUIDs are generated randomly.
39. (Previously Presented) The method of Claim 29, wherein SQUIDs are generated by hashing.
40. (Canceled)
41. (Currently Amended) The method of Claim [[40]] 29, comparing SQUIDs of two different pointers further comprising:
reordering instructions responsive to the comparison of SQUIDs.
42. (Currently Amended) The method of Claim [[40]] 29, comparing SQUIDs of two different pointers further comprising:

determining that the two pointers do not reference the same data object if the SQUIDs are different.

43. (Currently Amended) The method of Claim [[40]] 29, comparing SQUIDs of two different pointers further comprising:

determining that the two pointers reference the same data object if the SQUIDs are identical and address fields of the two pointers are identical.

44. (Currently Amended) The method of Claim [[40]] 29, comparing SQUIDs of two different pointers wherein each pointer address field comprises a base address and an offset, the method further comprising:

determining the two pointers do not reference identical locations within a referenced data object if the pointers' offsets are not identical.

45. (Currently Amended) The method of Claim [[40]] 29, comparing SQUIDs of two different pointers wherein a pointer is associated with a migration indicator field which indicates a number of migrations of the referenced data object by the time said pointer is created, the method further comprising:

determining that said two pointers do not reference the same data object if their associated migration indicators indicate identical numbers of migrations and their corresponding addresses are different.

46. (Previously Presented) The method of Claim 45, wherein the migration indicator comprises one bit.
47. (Previously Presented) The method of Claim 29, wherein at least one pointer is a guarded pointer.
48. (Previously Presented) The method of Claim 29, wherein the SQUID is implemented by hardware.

49. (Previously Presented) The method of Claim 29, wherein the SQUID is implemented by software.
50. (Previously Presented) The method of Claim 29, wherein a pointer contains its associated SQUID.
51. (Previously Presented) The method of Claim 29, further comprising:
maintaining a SQUID cache for storing SQUIDS of recently-used pointers.
52. (Original) A method for processing programs, the method comprising:
storing data objects in memory;
generating pointers to the stored data objects;
associating migration indicators with the pointers, a migration indicator indicating a number of migrations of a data object referenced by an associated pointer prior to said associated pointer being created; and
comparing two pointers and determining that said two pointers do not reference the same data object if their associated migration indicators indicate identical numbers of migrations and their corresponding addresses are different.
53. (Previously Presented) The method of Claim 52, wherein the migration indicator comprises one bit.
54. (Previously Presented) The method of Claim 52, wherein the migration indicator comprises multiple bits.
55. (Previously Presented) The method of Claim 52, wherein migration indicators are implemented by hardware.
56. (Previously Presented) The method of Claim 52, wherein the migration indicators are implemented by software.

57. (Currently Amended) A data processing system for processing programs, the system comprising:
means for storing data objects in a memory, the data objects being referenced by pointers; ~~[[and]]~~
means for generating a short-quasi-unique-identifier (SQUID) and assigning the SQUID to a data object stored in the memory segment, pointers to the data object containing the data object's assigned SQUID; and
means for comparing SQUIDs of two different pointers.
58. (Canceled)
59. (Original) A data processing system for processing programs, the method comprising:
means for storing data objects in memory;
means for generating pointers to the stored data objects;
means for associating migration indicators with the pointers, a migration indicator indicating a number of migrations of a data object referenced by an associated pointer prior to said associated pointer being created; and
means for comparing two pointers and determining that said two pointers do not reference the same data object if their associated migration indicators indicate identical numbers of migrations and their corresponding addresses are different.
60. (Currently Amended) A computer program product for processing programs, the computer program product comprising a computer usable medium having computer readable code thereon, including program code which:
stores data objects in a memory, the data objects being referenced by pointers; ~~and~~
generates a short-quasi-unique-identifier (SQUID) and assigns the SQUID to a data object stored in the memory segment, pointers to the data object containing the data object's assigned SQUID; and
compares SQUIDs of two different pointers.

61. (Canceled)
62. (Currently Amended) A computer data signal ~~embodied in a carrier wave~~ for processing programs, comprising:
- a program code segment for storing data objects in a memory, the data objects being referenced by pointers; [[and]]
 - a program code segment for generating a short-quasi-unique-identifier (SQUID) and assigning the SQUID to a data object stored in the memory segment, pointers to the data object containing the data object's assigned SQUID; and
 - a program code segment for comparing SQUIDs of two different pointers.
63. (Canceled)
64. (Original) A computer program product for processing programs, the computer program product comprising a computer usable medium having computer readable code thereon, including program code which:
- stores data objects in memory;
 - generates pointers to the stored data objects;
 - associates migration indicators with the pointers, a migration indicator indicating a number of migrations of a data object referenced by an associated pointer prior to said associated pointer being created; and
 - compares two pointers, determining that said two pointers do not reference the same data object if their associated migration indicators indicate identical numbers of migrations and their corresponding addresses are different.
65. (Currently Amended) A computer data signal ~~embodied in a carrier wave~~ for processing programs, comprising:
- a program code segment for storing data objects in memory;
 - a program code segment for generating pointers to the stored data objects;

a program code segment for associating migration indicators with the pointers, a migration indicator indicating a number of migrations of a data object referenced by an associated pointer prior to said associated pointer being created; and

a program code segment for comparing two pointers, said segment determining that said two pointers do not reference the same data object if their associated migration indicators indicate identical numbers of migrations and their corresponding addresses are different.